UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/458,121	12/08/1999	GAL MOAS	042390.P7162	8466

8791 7590 08/11/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 08/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/458,121	MOAS ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 May 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-11,13-20 and 22-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-11,13-20 and 22-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 5/28/06.

As indicated in Applicant's response, claims 1, 4-5, 10-11, 13-14, 19, 20 and 22 have been amended. Claims 1-2, 4-11, 13-20, 22-25 are pending in the office action.

Specification

2. The disclosure is objected to because of the following informalities: the term 'then' in the 2nd paragraph (1st line) of page 10 as in 'rather then at each' needs to be corrected.

Appropriate correction is required.

Claim Objections

3. Claims 1, 10 and 19 are objected to because of the following informalities:

Claims 1, 10 and 19 recite 'whether resource requirements of the first processor have been exceeded' (line 7). Based on the context set forth in the disclosure, only the resources needed at the beginning of a block are checked for their availability such that only the resource requirements from the instructions in the block of code are looked upon (Specifications pg. 9, bottom-pg. 10, top) to effectuate availability checking of resources. Normally, requirements set constraints and constraints include limits that are not to be exceeded, and based on the above part of the Specifications, no explicit association is depicted between setting of any constraints or limits (by a code instruction) and any resource requirements of any first processor (Note: the term processor is never mentioned in the paragraph) so that these would be exceeded. In short, as the Specifications has it that only the limits set by the requirements of the instructions that are checked, it would be improper to recite that the resource requirements (of a first processor) are

exceeded, lest that this would be leading to a 35 USC 112 type of rejection because of lack of description.

Claims 1, 10, 19 recite 'inserting a ... instruction ... at *a* start of a block of code *to run* on a first processor'; and according to the specifications (Background, pg. 2-3), the limitation is not really conveying the disclosure. According to which, the code is designed/written to be run on an old architecture and an emulation process is implemented to ensure that the old architecture code does not have any resource conflicts via implementation of non-intrusive code insertion using a newer architecture for emulating that code. One correction should be like 'at *the* start of a block of code *designed to be run* on *(* or *written for)* a first processor'.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1-2, 4-11, 13-20, 22-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

> The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

Specifically, claim 1 recites inserting a single instruction in a block of code; emulating the block of code on a second processor; and using said first instruction to monitor resources so

that *if* (emphasis added) the resource requirements have been exceeded, modifying allocation of the resources of the second processor. That is, the emulation a block of code is founded upon using an instruction at the start of the block to perform some reallocation of resources according to some requirements; and this is viable only to the extent that *if* said requirements are being exceeded or seriously endangered. The claim does not provide alternate teaching as to what would be performed in case no requirements conflict incur. Absent such alternate eventuality, one skill in the art would not be able to construe the realization of an action (for the emulation context involving 2 processors, a start instruction and some requirements) in order for what is being claimed to reasonably yield a concrete, useful and tangible practical result. From the Specifications, the advantage lies in the using of special code instructions before the emulation of any code block would be taken; and if resources are available according to some checking (via means of such instruction), the emulation would go on. Clearly, the whole emulation novel aspect or usefulness is conceived upon the running of the instruction being inserted; and the claim as recited leads to the eventuality where no action would be taken thus does not enable the usefulness thereof via the subsequent modifying the resources allocation as disclosed.

Absent any alternative to convey that the invention can generate a concrete, useful and tangible result, the claim amounts to a non practical idea, hence is rejected for leading to a non-statutory subject matter.

Claims 10 and 19 are rejected likewise; and 2, 4-9, 11, 13-18, 20, 22-25 are also rejected for not remedying to the deficiency of not providing sufficient alternate way to carry out the invention to reach a useful result.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7. Claims 1-2, 4-11, 13-20, 22-25 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

The claims 1 and 10 recites 'emulating the block ... on a second processor;' and 'using the single instruction to monitor the resources ... have been exceeded' and '... allocating of the resources ... according to the resource requirements ... first processor'. From the disclosure, the insertion of code to monitor resources availability of code blocks of the emulating process is implemented to support the emulation of the old architecture via new architecture. According to the steps of the claim, such emulation is one step; and the use of the instruction for monitoring leading to modifying step appear to be subsequent steps that do not fall inside the above emulation process, but rather after what appear to be when the emulation step has been completed. One skill in the art would not be able to make use of the invention when the steps as claimed do not really reflect the time frame in which the resources availability monitoring and adjusting has been implemented in a same time sequence as that in the disclosure. That is, one skill in the art would not be able to reconstruct the invention if the claimed steps are not supported by the Specifications, i.e. the modifying step cannot be implemented after the emulation would be over. The steps as recited amount to subject matter that was not described in

the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The interpretation of the claim will be based on broad interpretation that the recited steps are loosely interrelated, unassociated or not bound in a specific time frame.

Claims 2, 4-9, 11, 13-18, 20, 22-25 are also rejected for not remedying to the above deficiencies; and the lack of clarity would be treated as though the emulation is not encompassing the above steps.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-2, 5, 10-11, 14, 19-20, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Long et al., USPN: 5,835,958, and further in view of Admitted Prior Art (Application Specs: Background, pg. 6, pg. 9 – hereinafter APA)

As per claim 1, Long discloses a method comprising:

inserting a single instruction at the start of a block of code (step 204 – Fig. 2a; *wrapper* - col. 5, lines 8-17 – Note: a call to a stack-checking function reads on a single instruction) to determine if resources of a processor are available for the block of code, wherein the block of code includes multiple instructions (step 206 – Fig. 2A – Note: a function being called reads on

multiple instructions); and using the instruction to monitor resources to determine whether resources requirements have been exceeded (see above);

if the resources requirements have been exceeded, modifying allocation of the resources (e.g. Fig. 2A; col. 5, lines 18-28; col. 6, line 64 to col. 7, line 28; *trampoline function* - col. 7, line 66 to col. 8, line 27) according to resources requirements.

But Long does not explicitly disclose inserted instruction in block of code to be run by a first processor; emulating the block of code by a second processor and monitoring using the single instruction resources of the second processor to determine whether resources consumption of the first processor be exceeded and modifying based thereupon to modifying allocation of said resources of the second processor. Long discloses using single instruction to readjust stack resources on a processor so to enable multi-platform applicability of runtime stack reallocation (see Long BACKGROUND, col. 1-2); thus obviate the need would require committing a particular compiler for a given architecture. Analogous to Long's approach to check runtime resources applicable to more than one platform, APA has applied code headers to check of stack resources in platform emulation. That is, using of emulation of code by one platform (emulating platform, or second processor) to check conflicts on target platform (or first processor) via code included in headers of block is disclosed in APA (e.g. *older processor, newer processor, older processor architecture* -- Application Specifications, pg. 2-3, pg. 6). Hence according to the as-needed basis of code insertion for compilers to provide a means to check any runtime platform by Long, it would have been obvious for one skill in the art at the time the invention was made to provide the code checking implementation as taught by Long to the emulation as purported by the APA in support for emulation different architectures as mentioned by Long which also taught

by APA. One would be motivated to do so because this would obviate unnecessary code intrusion or alleviating runtime code inefficiency as endeavored by Long, i.e. redundant epilogue being pre-integrated and imparted by compilers can be obviated (see Long: col. 2); and further this would enable more efficient way of assessing of newer architectures in view of older architecture (see APA, pg. 2) without dedicating architecture-specific compilers as being concerned by Long.

As per claim 2, Long discloses determining a set of available resources that will be available after said block of code has executed (e.g. Fig. 4; *epilogue... new stack chunk to be executed.. re-lock... reflector frame* – col. 9, lines 10-20; col. 10, lines 15-21).

As per claim 5, Long discloses availability determined at runtime (Fig. 2A-B – Note: usage and manipulation of data already stored in a stack reads on dynamic process).

As per claim 10, Long discloses a computer-readable medium having stored thereon a set of instructions to monitor processor resources (e.g. Fig. 5), said set of instruction, which when executed by a processor, cause said processor to perform a method comprising:

inserting (a single instruction at a start of); using the instruction to monitor (resources ... to determine if resources requirements ... exceeded); and

if the resource requirements are exceeded, modifying allocation (... according to resource requirements).

All of these limitations have been addressed in claim 1; hence this claim is rejected based on the corresponding rationale as set forth therein.

But Long does not explicitly disclose inserted instruction in block of code to be run by a first processor; emulating the block of code by a second processor and monitoring using the

single instruction resources of the second processor to determine whether resources consumption of the first processor be exceeded and modifying based thereupon to modifying allocation of said resources of the second processor. However, this limitation has been addressed in claim 1.

As per claims 11 and 14, these claims are the computer-readable medium or apparatus claims corresponding to method claims 2 and 5, respectively, hence are rejected herein with the same reasons as set forth above.

As per claim 19, Benson discloses a computer readable medium having a first set of instructions (e.g. Fig. 5), which when executed, generate a second set of instructions through a binary translation process, the second set of instructions (e.g. Fig. 2A-B – Note: verifying code invocation reads on second set of instructions) when executed cause the processor to perform a method comprising the steps of:

inserting (a single instruction at a start of...); using the single instruction to (... resources ... to determine if resources requirements ... exceeded); and

if the resource requirements are exceeded, modifying allocation (... according to resource requirements).

All of these limitations have been addressed in claim 1; hence this claim is rejected based on the corresponding rationale as set forth therein.

But Long does not explicitly disclose inserted instruction in block of code to be run by a first processor; emulating the block of code by a second processor and monitoring using the single instruction resources of the second processor to determine whether resources consumption of the first processor be exceeded and modifying based thereupon to modifying allocation of said resources of the second processor. However, this limitation has been addressed in claim 1.

As per claims 20 and 22, this claim corresponds to claim 2 and 5; hence is rejected with the corresponding rejection as set forth therein.

10. Claims 4, 6-9, 13, 15-18, and 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Long et al., USPN: 5,835,958, and of Admitted Prior Art (APA) and further in view of Yellin et al., USPN: 5,740,441(hereinafter Yellin).

As per claim 4, Long does not explicitly disclose the availability of the processor resources is determined at a compile time; but Long teach pre-execution stack-oriented verifying method that is desired to grow dynamically on a as-needed basis in an platform-independent software execution environment (see col. 1, line 50 to col. 2, line 31). Yellin, in a method to execute Java bytecodes which is analogous to the platform independent stack-based endeavor by Long, discloses verifier code (see *jsr() call, SnapShot data, program Verifier*) for checking stack data and a class loader for a interpreter/compiler Java runtime machine in which bytescode are interpreted or dynamically translated for execution (e.g. Fig. 4a-c; *compiled into bytecodes* - col. 12, line 61 to col. 13, line 59). In view of the desirability of using non-intrusive insertion means for pre-checking of architecture usage as by APA and the runtime approach by Long as set forth in claim 1, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide Long/APA with Yellin's pre-execution verification engine so that it perform the stack resources checking in a Java interpreter/compiler (as taught by Yellin). Based on the well-known concept of a Just-in-Time compiler similar to Yellin's stack-based pre-runtime method, one would be motivated to do the modification to Long because it would provide Long with the platform flexibility and possibility to obviate unnecessary recompiling as

desired by Long dynamic stack growing method, which is shown via Yellin's stack-based verification process working on the multi-platform portable bytecodes.

As per claim 6, Long does not explicitly disclose signaling an error message if the resources of the processor needed for the block of code are not available; and in response to the error message branching to a fault handler routine. But in view of the limited resources on a given architecture stack, memory availability is not unlimited; and according to a network connected runtime execution by Long, when memory conflict occurs, an exception as a error to alert the system would be suggested (see memory allocation errors - col. 10, line 48 to col. 10, line 6). Based on the Java interpreter/compiler by Yellin, memory violation at runtime can be thrown as a message to be handled via exception handler (e.g. Yellin: Appendix 1, col. 14 lines 58-65; step *440* Fig. 4B; step *454* Fig. 4C). In view of the bytecodes inter-platform operability and interpreter Java engine over network connected machines and the rationale as set forth in claim 4, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide Long with error handling message as taught by Yellin during execution of network bytecodes when certain memory resources are conflicting with the demand of the running code, when all the dynamic allocation have been exhausted because only via exception being thrown (i.e. as a error message) as above shown by Yellin would Long's runtime environment be prevented from incurring more memory catastrophic failures.

As per claim 7, this claim subject matter falls under the ambit of error message and handler by a processor to handle the memory conflict exception in claim 6; hence is rejected using the rationale as set forth therein.

As per claim 8, there is no explicit teaching by Long on a bit vector. However, Yellin discloses the stack resources are represented by a bit vector (col. 7, lines 25-55) while stack memory by Long is managed and observed for allocation as chunks via a *stack chunk context structure* that indicates how stack resources are being used (Fig. 2; *stack reflector* - col. 7, line 9-65). Arranging a structure of stack chunk to reflect the usage condition of the stack resources by Long can be viewed as a vector indicating status of stack chunk usage as purported by Yellin. In case the target code by Long is a bytecodes for an interpreter in a Java runtime environment as shown by Yellin (see TABLE 1, TABLE 2 col. 15-21) as set forth via the rationale of claim 4, it would have been obvious for one of ordinary skill in the art at the time the invention was made to make the stack chunk structure by Long so that it is implemented as a JSR bit vector as by Yellin, because of the JSR bit vector is a Java code handy to be use to support the snapshot of the stack resources as purported by Yellin, and this is exactly what the stack reflector process used by Long is intended to do; making the bit vector in Java useful because the JSR call is written in the very same code as the target language.

As per claim 9, Yellin disclose a JSR call at pre-execution of a bytecode hence Long combined with Yellin by virtue of claim 8, disclose wherein said bit vector is generated dynamically.

As per claims 13, 15-18, these claims are the computer-readable medium or apparatus claims corresponding to method claims 4, 6-9, respectively, hence are rejected herein with the same reasons as set forth above.

As per claims 23-25, these claims are the computer-readable medium or apparatus claims corresponding to method claims 6-8, respectively, hence are rejected herein with the same reasons as set forth above.

Response to Arguments

11. Applicant's arguments filed 5/28/2005 have been fully considered but they are mostly moot in view of the new grounds of rejection or non persuasive. Following are the Examiner's observations in regard thereto.

(A) Applicants have submitted that nowhere does Long disclose emulating code or determining if resource requirements are exceeded (Appl. Rmrks pg. 8 bottom). The intended use of resources checking by Long in a emulation has been set forth in the Office Action using a combination of teaching; making an intended use of a methodology (applying Long's method in a emulation application) much less of a novel weight than as it was intended to be as from the Applicant's remark. The rejection now uses a combination of teaching and one reference cannot be subject to rebut as a stand-alone reference. Besides, the reciting of 'resources requirements ... exceeded' in the claim is construed as improper term usage as set forth in the Office Action.

(B) Applicants have submitted that Yellin does not disclose or suggest emulating of a block of code (Appl. Rmrks pg. 9, 2nd 3rd para). The intended use of resources checking by Long in a emulation has been set forth in the Office Action using a combination of teaching using Yellin for another purpose; making an intended use such as an emulation process less of a novelty as it is intended to be as from the Applicant's remark. The rejection now uses a combination of teachings and one reference cannot be rebutted as a stand-alone reference as if it were applied in an anticipating type of rejection. The arguments as a whole are not persuasive given the amount

of deficiencies of the claim language as set forth in the Office Action, and in view of the new grounds of rejection.

The new grounds of rejection necessitated by the new amendments will stand.

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
July 25, 2006

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100